

Error Detection in SMBus

Hitachi
 NEC Moli Energy
 Mitsubishi System LSI Design
 Fujitsu

1. Background

SMBus specification does not support any error detection mechanism. The SMBus host needs some extra operations for increasing the data reliability. For example, some SMBus host reads the battery data several times and compares these data to make sure that the data it received are correct. To improve the reliability, we suggest to support an error check function as follows.

2. Error Sources

- Electrical and circuit noise (e.g. connector chattering noise)
- Data error from some problem in a sender device
- Firmware implementation

3. Check Code

- LRC (Longitudinal Redundancy Check)
- CRC (Cyclic Redundancy Check)

CRC performs better than LRC, but its implementation is more complex and gives more big impact to the devices. Therefore, LRC is better solution for SMBus spec Rev.1.1.

4. Implementation

- LRC implementation will add one more byte for error detection code to SMBus protocol
- LRC should be an optional function in SMBus spec Rev 1.1
- Both devices that implement LRC and Rev 1.0 should be able to co-exist
- Must keep backward compatibility

4-1. Support Protocol

LRC should be supported as a following table.

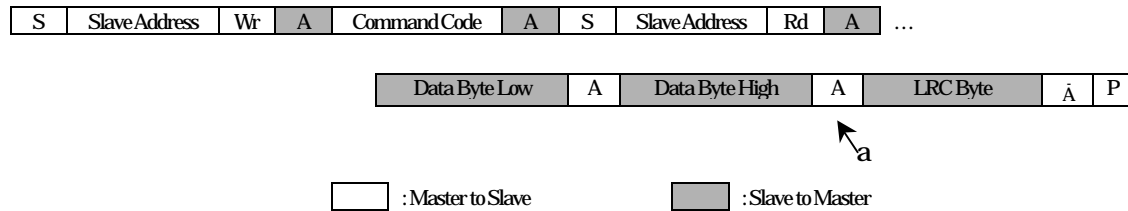
Protocol	LRC function	Note
Quick Command	No	This command doesn't have data region
Send Byte	Yes	
Receive Byte	Yes	
Write Byte/Word	Yes	
Read Byte/Word	Yes	
Process Call	Yes	
Block Read	Yes	
Block Write	Yes	
Host Communicating	No	Embedded controller does not have any readable register from other devices
SMB Alert	No	This command doesn't have data region

* The broadcast from the batteries does not need to support LRC Byte.

4.2. LRC Byte

LRC Byte is attached behind last data byte.

[Ex.] Read Word Protocol

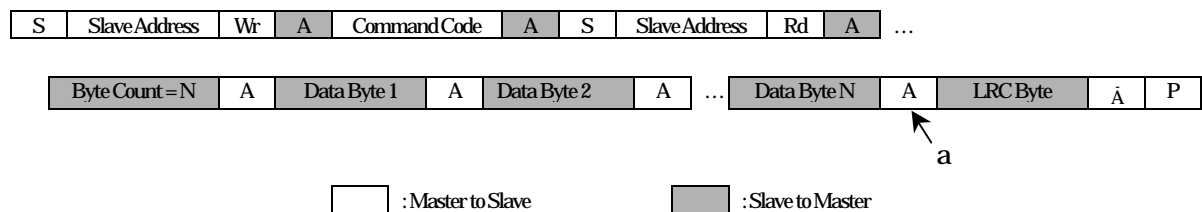


If master device doesn't support LRC function, Master should send NACK and STOP CONDITION to a slave device at the point "a" (same as a normal protocol) and the slave device clears its LRC Byte buffer.

LRC Byte can be calculated with the following formula,

$$[\text{Slave Address} + \text{Wr}] \text{ XOR } [\text{Command Code}] \text{ XOR } [\text{Slave Address} + \text{Rd}] \text{ XOR } [\text{Data Low}] \text{ XOR } [\text{Data High}]$$

[Ex.] Block Read Protocol

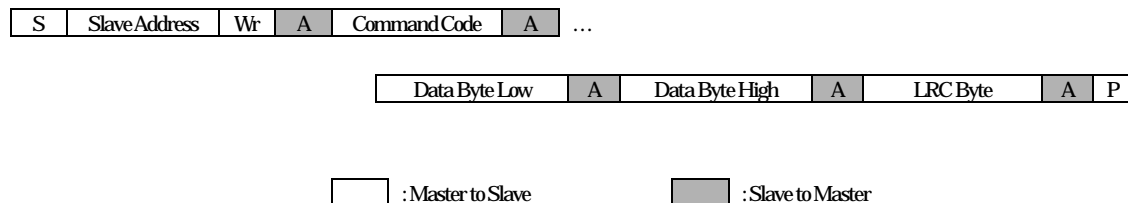


If master device doesn't support LRC function, Master should send STOP CONDITION to a slave device at point "a" (same as a normal protocol) and the slave device clears its LRC Byte buffer.

LRC Byte can be calculated with a following formula,

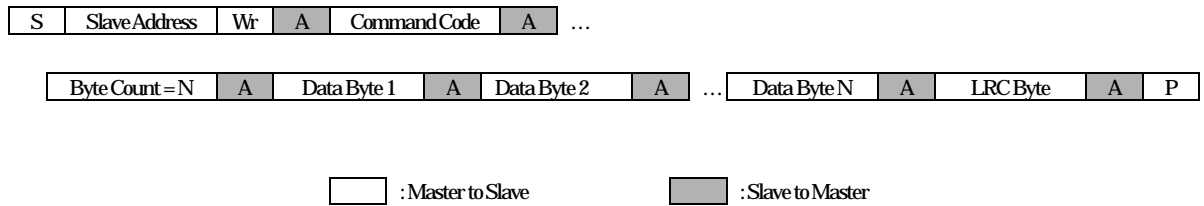
$$[\text{Slave Address} + \text{Wr}] \text{ XOR } [\text{Command Code}] \text{ XOR } [\text{Slave Address} + \text{Rd}] \text{ XOR } [\text{Data 1}] \text{ XOR } [\text{Data 2}] \text{ XOR } \dots [\text{Data N}]$$

[Ex.] Write Word Protocol



LRC Byte can be calculated with the following formula,

$$[\text{Slave Address} + \text{Wr}] \text{ XOR } [\text{Command Code}] \text{ XOR } [\text{Data Low}] \text{ XOR } [\text{Data High}]$$

[Ex.] Block Write Protocol

LRC Byte can be calculated with a following formula,

[Slave Address] XOR [Command Code] XOR [Byte Count] XOR [Data 1] XOR [Data 2] XOR ... [Data N]

4.3. LRC Mode Register

Supporting LRC devices should support an “LRC Support” bit, and the bit indicates that the device supports LRC function. Smart Battery Devices set these bits as a following table.

Device Name	Command Code	Bit
Smart Battery Charger	0x12 (ChargerMode)	Bit 4
Smart Battery Selector	0x04 (SelectorInfo)	Bit 14
Smart Battery	0x03 (BatteryMode)	Bit 10

Value	Meaning
0	This device supports LRC function
1	This device does not support LRC function

The LRC function supported master device must read LRC Mode Register in the target slave devices before issuing the write command.

5. Device Impact

LRC implementation gives an impact to the following devices.

- SMBus Host (Embedded Controller)
- Smart Battery
- Smart Battery Charger
- Smart Battery Selector